Compress all source code files into a .zip file.

* A report in Microsoft Word (or PDF), including:
  + Page URL.
  + Describe the testing environment including device model, browser type, and resolutions (viewport size).
  + Screenshots with explanations. Make sure your screenshots are clearly shown and without cropping.

Page URL : file:///Users/mouniatouil/Desktop/Foundation%20Database%20/assignmentafterupdates/index.html#

Tested the page in three different views :

1. Device Model : Iphone SE ( image 1 and image 2)

viewport size : (375 X 667)

Testing Environment : Google Chrome

Screenshots : ( image 1 and image 2)

1. Device Model : Ipad Mini

viewport size : (768 X 1024)

Testing Environment : Google Chrome

Screenshots : ( image 3 and image 4)

1. Device Model : Desktop

viewport size : (1533 X 871)

Testing Environment : Google Chrome

Screenshots : ( image 5 and image 6)

Patterns applied :

**1. Flexible Grid Layout or Luke’s wroblewski’s pattern**

* Instead of a **fixed width (1200px)**, the layout now uses **percentage-based widths** so it adjusts to the screen size.
* Ensure content resizes appropriately based on screen size.
* Luke’s pattern recommends using mobile first strategy which is achieved using width as percentage.

Example:

* #container { width: 100%; max-width: 1200px; margin: 0 auto; padding: 20px; display: flex; flex-wrap: wrap; justify-content: space-between; }

**2. Flexible Layouts :**

* The container and header use **display: flex** and flex-wrap:wrap to create a flexible layout that adjusts as the viewport size changes.

For example: the flex and flex wrap in the container and header.

#container { width: 100%; max-width: 1200px; margin: 0 auto; padding: 20px; display: flex; flex-wrap: wrap; justify-content: space-between; }

header { overflow: hidden; clear: both; margin-bottom: 30px; display: flex; flex-wrap: wrap; justify-content: space-between; align-items: center; width: 100%; }

**3. Media Queries for Different Viewports**

Two media queries are used to adjust the layout for different screen sizes (max-width of 1024px and 600px):

* On screens less than 1024px, the layout becomes more vertical (flex-direction: colums), the content and sidebar take up 100% width, and navigation changes to a column format.
* On screens less than 600px, the font sizes decrease for better readability on smaller devices, and the layout adapts to a fully stacked format for all sections.

Example

@media (max-width: 1024px) { header { flex-direction: column; text-align: center; }

header nav ul { flex-direction: column; }

#content { width: 100%; }

#sidebar { width: 100%; max-width: none; }

}

@media (max-width: 600px) {

header hgroup h1 { font-size: 30px; }

header hgroup h2 { font-size: 20px; }

header nav ul { flex-direction: column; }

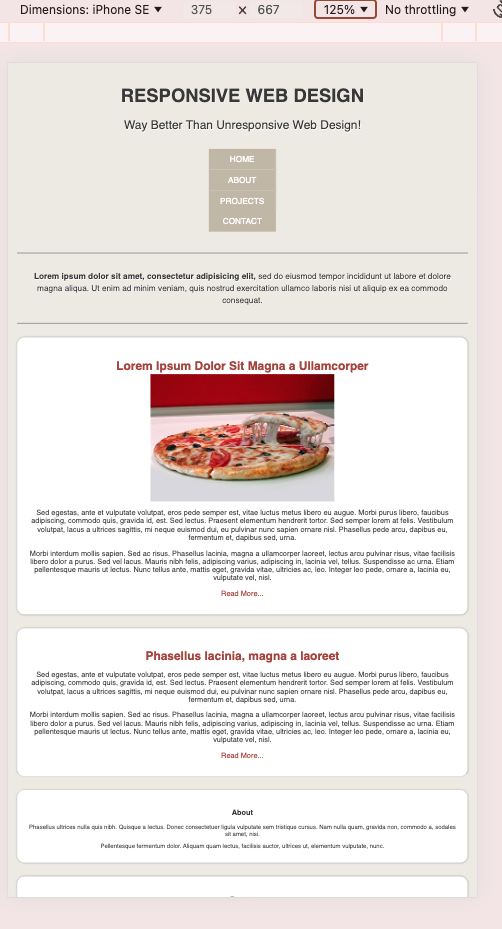
header nav ul li { width: 100%; text-align: center; border: none; }

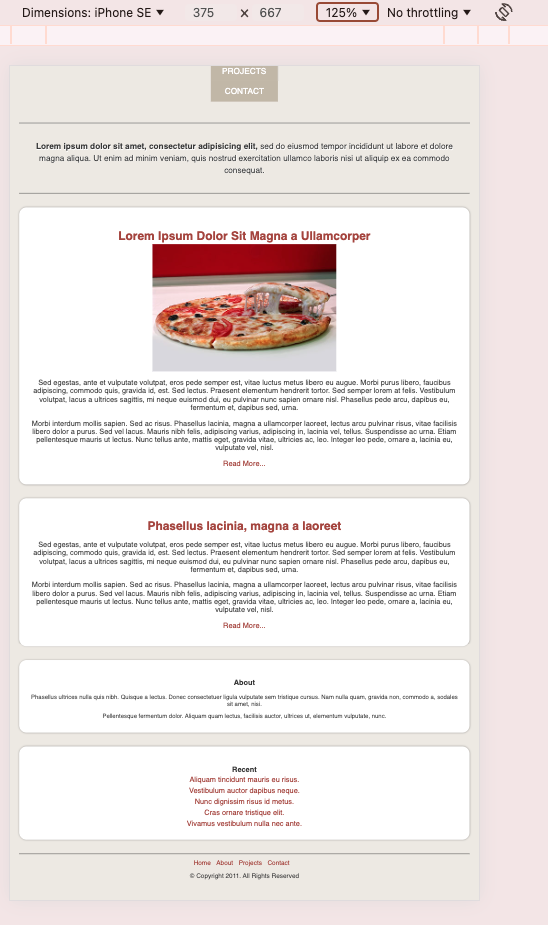
#content { width: 100%; }

#sidebar { width: 100%; }

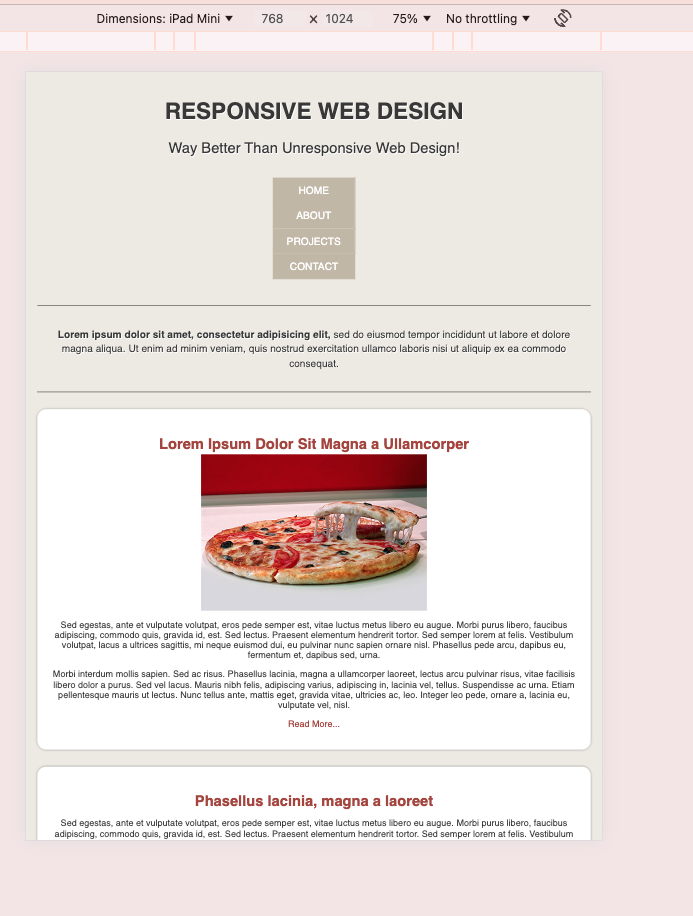
#content article { padding: 10px; }

}

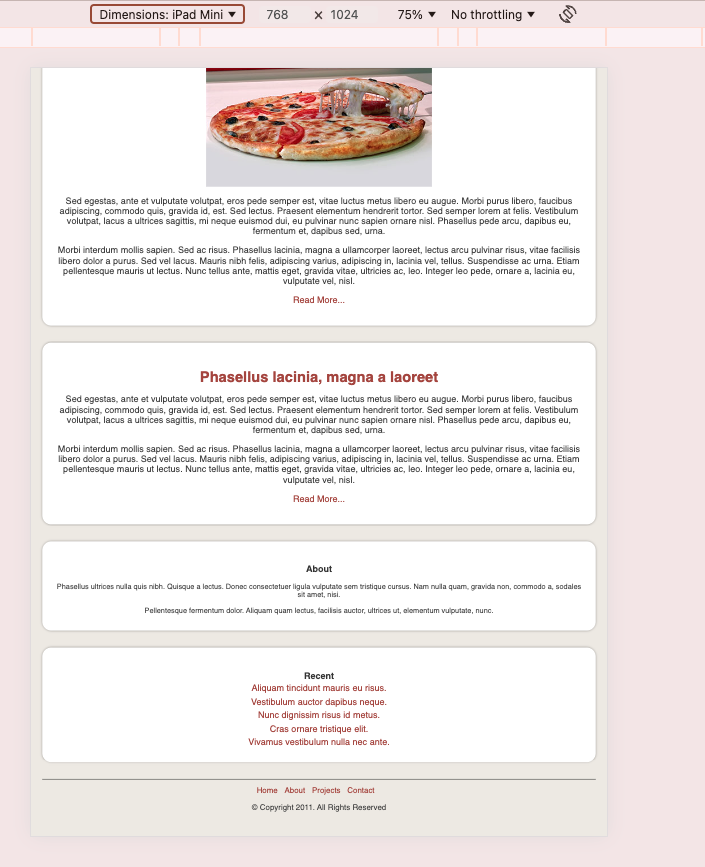
**Image 1 : Iphone SE screenshot 1 showing the top before scroll (**Navigation **becomes a vertical stack,** Sidebar **is below the content,** Text & images **resize automaticall)y**

**Image 2 : Iphone SE screenshot 2 showing the bottom after scroll** 

**Image 3 : Ipad Mini view screenshot 1 before scroll**

Sidebar **drops below** content

**Image 4 : Ipad Mini view screenshot 2 after scroll**

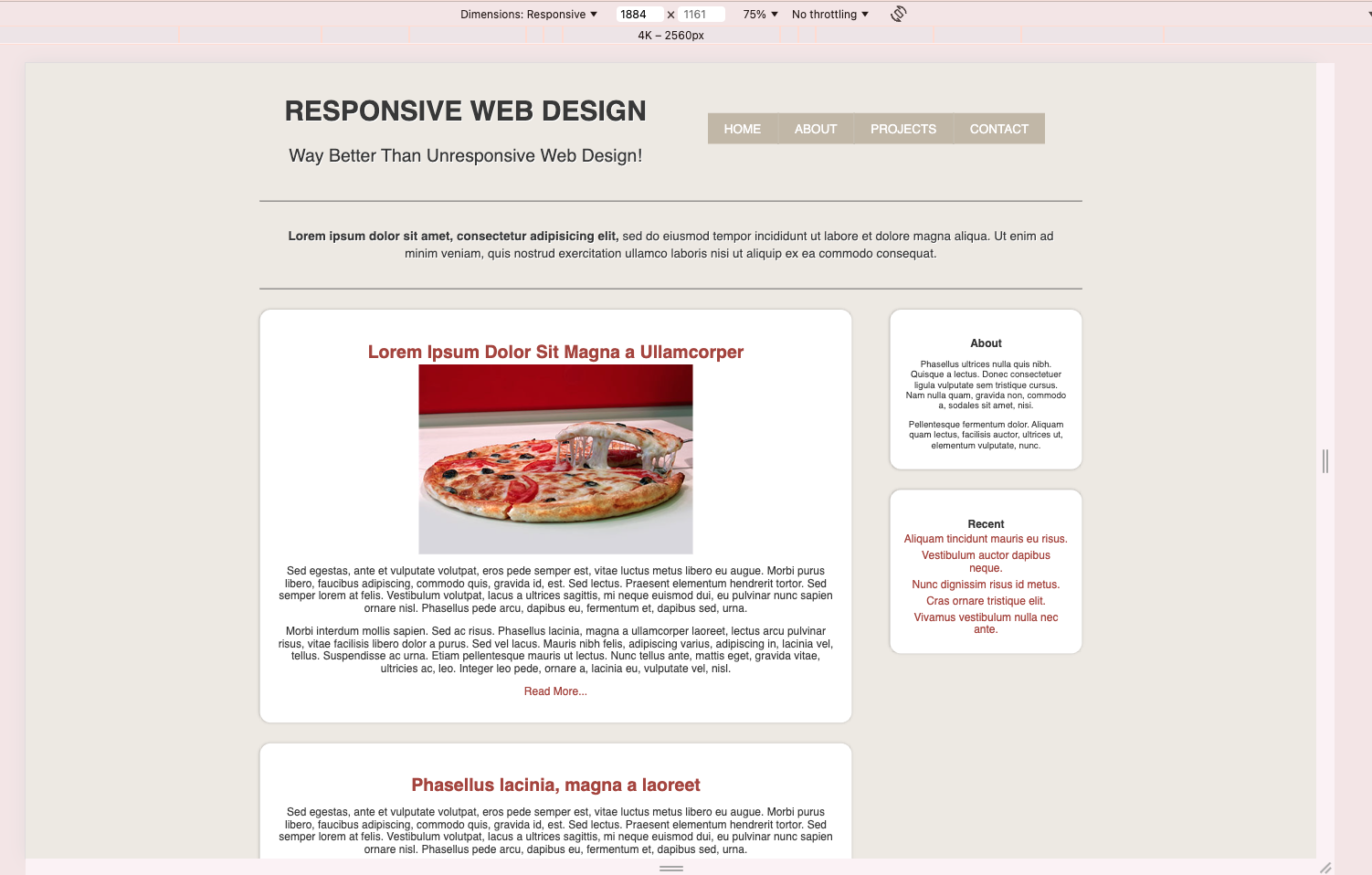
****

**Desktop view Explanation :**

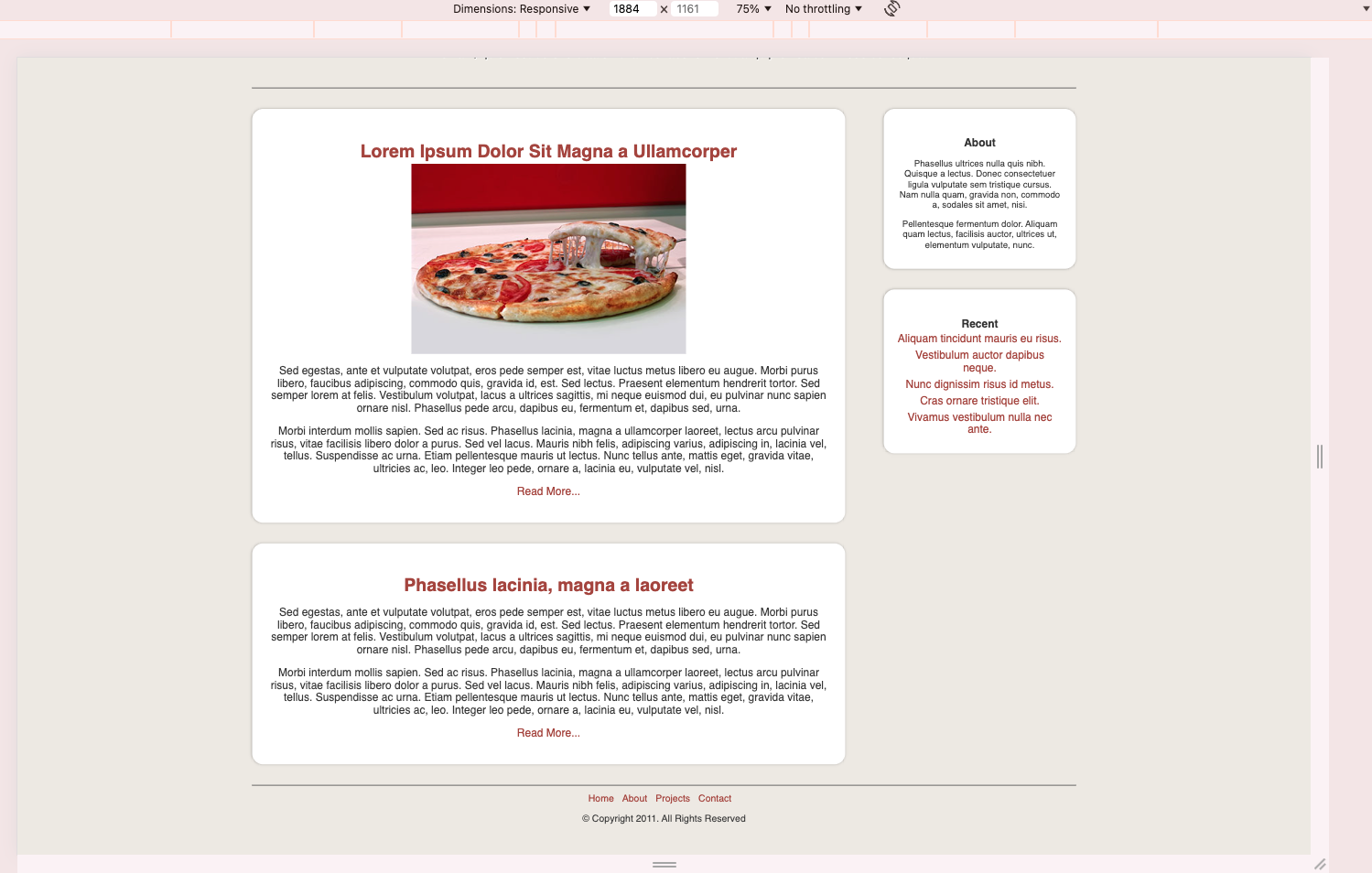
**Sidebar is on the right of the content**

**No elements overlap**

**Image 5 : Desktop view screenshot 1 before scroll**

****

**Image 6 : Desktop view screenshot 2 after scroll**

****